

INTT Weekly Meeting

Joseph Bertaux

Purdue University

June 5, 2024



- Had to make drastic changes after arriving two weeks ago per requests of Chris and Jamie
- Needed to simplify the implementation
 - Removed TExec functionality (no pop-up/sub windows)
 - Rewrote all methods
 - BCO difference method shown in previous meeting

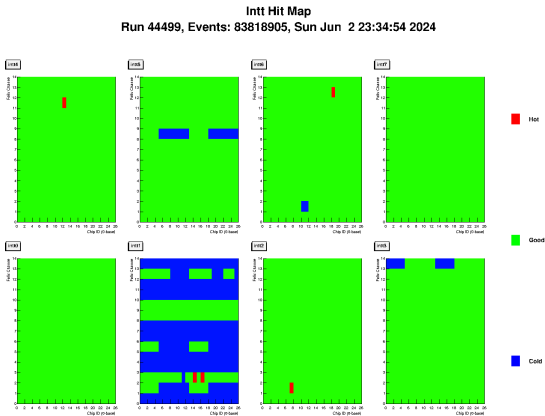


Figure: OnlMon option showing hitmap. Hot/Cold/Dead are defined by hits/pmonitor event (10^{-4} and 10^{-2}). intt1 looks cold, but this is because the two hot channels it has are so hot it is saturating the DAQ.

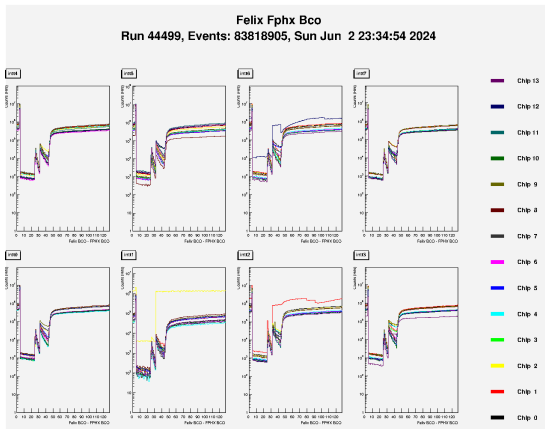


Figure: OnlMon option showing BCO difference distribution. The Felix channels which had hot channels are distinct from other distributions (intt1 Felix channel 2 and intt2 Felix channel 1, for example)

Follow through with some proposed changes to the software:

- Maintain the Felix-Ladder mapping in CDB
- Base class for calibrations that are loaded from the CDB
 - Common handle `Load(std::string const&)` that can load files or payloads
 - Simplify method handles to `InttCombinedRawDataDecoder`
- Revisiting per-run calibrations (originally due to Jaein)
 - For hot channel classification, normalize using survey geometry
 - Allows hitrates to be compared for different layers
 - Check state of calibrations directory

Common handle for loading that switches on the extension of the given argument:

```
int InttLoadable::Load(std::string const& name)
{
    // guard clauses on name
    // ...

    std::string filename = name.find(".root") != std::string::npos ?
        name : CDBInterface::instance()->getUrl(name);

    // gaurd clauses on filename
    // ...

    // normal loading
    CDBTTree cdbttree(filename);
    cdbttree.LoadCalibrations();

    // call protected member function
    // to actually intialize with valid CDBTTree instance
    // ...
}
```

This base has a handle `IsLoaded` allowing users to check (success is also indicated by the return value of `Load`)

Decoder handles can be simplified, for example the BCO map:

```
// in InttCombinedRawDataDecoder.h
public:
    // Handles for each calibration; BCO map for example:
    int LoadBcoMap(std::string const& name = "INTT_BCOMAP")
        {return m_bcomap.Load(name);}
    //...
private:
    //...
    InttBCOMap m_bcomap;
    //...

// in InttCombinedRawDataDecoder::InitRun()
if (!m_bcomap.IsLoaded())
{
    if (LoadBcoMap() && Verbosity())
    {
        std::cout << __PRETTY_FUNCTION__ << "\n"
        << "\tCould not load m_bcomap\n"
        << "\tNo-time-window-filetering-will-be-performed\n"
        << "\tDecoder-will-still-run" << std::endl;
    }
}
```

- I wanted to implement these sooner
- However, I was on shift in March (for 4 weeks), and Hao-Ren was making commits to implement the survey geometry
- They were not merged during this time
- I've just finished my recent shift and it's a good time to revisit it, I have a draft PR in progress:
- <https://github.com/sPHENIX-Collaboration/coresoftware/pull/2816>

- Jaein has committed modules to coresoftware/calibrations
- I'm not sure these are being used for production
- Chris asked me to check on their status